Page 4, line 10

B2 The zoom on the second display can be filtered to avoid pixelation (i.e. an image not illustrating coarse pixels) or unfiltered (i.e. an image illustrating coarse pixels) to facilitate pixel by pixel viewing and editing. User input can also define a user's choice of filtering or non-filtering.

Page 6, line 24

B3 Figure 1 shows a high level block diagram of the preferred embodiment. Two CRTC's 11 and 12 are capable of fetching one or more display surfaces from a single frame buffer memory (50) which can be SGRAM, SDRAM, or any other type of Random Access Memory (RAM). Each CRTC may also contain one or more backend scalers that allows the input surfaces to be re-scaled. While, within the context of the present invention, each controller 11 and 12 does not need to access more than one surface, greater image processing and display ability may be provided when multiple surfaces can be accessed by each controller.

Page 7, line 28

B4 Figure 2 shows a flow chart for the embodiment illustrated in Figs. 4 and 5. When the end user enables the zoom using a hotkey (H1) or the like in step 100, the software allows the user to select a rectangular window from the primary display in step 101. Once the zoom operation is enabled, the hotkey is also detected (step 100') to determine if the zoom operation should be disabled (step 113). One example of this could be that the user holds down the mouse key at which point the coordinates of one corner of the zoom window are determined. The user then drags the mouse while holding down the key and stops at the corner diagonally opposite the first one to specify the rectangle and lets go of the key. At this point the coordinates of the corner diagonally opposite the first one are determined and this information is enough to specify the size and location of the zoom window. Of course, there are

B4

many other ways to determine this rectangular area without departing from the spirit and scope of the invention. The coordinates of the zoom window (including address in memory) are thus stored. As illustrated in Fig. 2, the coordinates of the zoom window are sent to the display driver in step 102.

Page 8, line 10

sub B3

B5

The resolution of the destination can be either automatically calculated or user defined (step 103). When it is user defined, the software uses this resolution (step 104). In the preferred embodiment, it is automatic, it could be chosen in a variety of ways ranging from (but not limited to) the closest standard resolution (to the resolution of the zoom window) to the largest resolution possible etc. These resolutions determination options can also be specified by the user. Once the destination resolution is chosen, the scaling factor is determined (step 106). This determination of the scaling factor is within the general knowledge of those in skilled in the art.

Page 9, line 10

B6

With knowledge of the destination resolution, a buffer of this resolution is reserved in memory (step 105) for the zoomed area (zoom buffer). As will be appreciated, multiple buffers can be allocated if double or triple buffering is desired and when multiple zoom windows are defined. The secondary CRTC is then programmed (step 107) to read from this zoom buffer (or set of zoom buffers). If the panning or mouse following feature is enabled (step 108) then the location of the zoom window is consistently updated (step 109), see Figure 3. In this embodiment, the zoom window can be locked to the movement of the mouse and the zoomed area is updated in real-time.

Page 11, line 12